Design Space Exploration of Vector Architectures for Multimedia Applications

氏名（本籍地）	GAO YE 高 也
学位の種類	博　士（情報科学）
学位記番号	情　博　第　563　号
学位授与年月日	平成２６年　３月２６日
学位授与の要件	学位規則第４条第１項該当
研究科、専攻	東北大学大学院情報科学研究科（博士課程）　情報基礎科学専攻
学位論文題目	Design Space Exploration of Vector Architectures for Multimedia Applications （マルチメディアアプリケーションのためのベクトルアーキテクチャ設計）
論文審査委員	（主査）東北大学教　授　小林 広明 東北大学教　授　亀山 充隆　東北大学教　授　青木 孝文 東北大学准教授　滝沢 寛之　東北大学准教授　江川 隆輔

論文内容の要旨

Chapter 1

People have never ceased to aspire to a higher quality of media services. In order to realize the high quality media service, next generation multimedia applications (MMAs) will process an unprecedented amount of media data in real time. Therefore, both a high computing performance and a high data transfer performance are required for the processors that execute the high quality MMAs so as to efficiently process media data. On the other hand, power budget of the processors that executes MMAs cannot be significantly improved in the desktop computers because of the limitations of sizes and performances of the power supply unit and cooling unit. Therefore, it is required for the processors to achieve a power-efficient execution for MMAs. In addition to power efficiency, processors to execute MMAs are also required to have a high programmability because more and more MMAs will be executed on one platform. Therefore, a processor that enables to accelerate a wide range of MMAs is required.

Nowadays, one of the most effective approaches to satisfying this requirement is to enhance general purposes processors (GPPs) with SIMD extensions. The SIMD extension could process multiple data by using one instruction. Such a mechanism enable to exploit data level parallelism (DLP) involved in MMAs to achieve a high power efficient execution on MMAs.

So far, the peak computing performance of the SIMD extensions increases significantly by improving the width of SIMD execution units. However, with the failure of Dennard scaling, straightforwardly increasing the width of SIMD execution units cannot be improve the power efficiency any more. It is necessary to consider a more efficient way to use the SIMD execution units. One approach is to deeply pipeline the SIMD execution units and use a large amount data to fill up the pipelines.

Indeed, there is such an architecture. That is the vector architecture. The vector architecture is usually used in high performance computing systems, or so called supercomputers. Since the applications in high performance computing domains usually have a long vector in their algorithms, the vector architecture usually employs large vector registers to store the long vectors and fill up the deeply pipelined vector functional units (VFUs) and vector load and store unit (VLSU). In this way, the stalls due to data dependency and latencies of VFUs and VLSU enable to be hidden, thus leading to a high efficient use of VFUs and VLSU. Since MMAs also contains a mass of data parallel processing, the vector architecture is also a promoting approach to improving the power efficiency of ISA extensions for MMAs. Therefore, the objective of this dissertation is to design a vector extension to acceleration a wide range of MMAs at high power efficiency.

However, the vector architecture still has its own inefficiencies for MMAs. The first problem is that conventional vector architecture cannot efficiently process short vectors, which are commonly involved in MMAs. The second problem is that the conventional multi-banked cache memory designed for vector architectures cannot achieve a high data transfer throughput at lower power consumption. The third problem is that the conventional vector architecture lacks the flexibility to match various hardware requirements from each MMA.

In order to resolve these problems, this dissertation proposes a media-oriented vector extension (MVPX).

MVPX employs an out-of-order vector processing mechanism and a multi-banked cache memory that associates single tag array with multiple data array in order to improve the power efficiency on short vector processing and data transfers, respectively. This dissertation also proposes an optimization method for MVPX in order to find the power efficient configuration for each MMA.

Chapter 2

Although the vector architecture shows a high potential on MMAs, it cannot efficiently process short vectors involved in MMAs. Most of the conventional vector architectures are designed and optimized for the applications in the high performance computing domains. Those vector architectures employ an in-order vector processing mechanism (IVPM), which easily causes pipeline stalls. The stalls would expose the memory access latency and the latencies of deeply pipelined VFUs. This method is effective for MMAs with long vectors because there are sufficient data to fill up the VLSU pipeline stages. However, when a MMA with short vectors is executed, the pipeline stalls would frequently occur, and thus expose the long access latency. The exposure causes the degradation of computational efficiency, leading to low power-efficient execution for MMAs with short vectors.

In order to reduce the pipeline stalls and hide the memory access latency, an out-of-order vector processing mechanism (OVPM) is proposed. OVPM can effectively schedule the vector instructions to hide pipeline stalls and memory access latency. By using OVPM, even though a vector instruction is waiting for its operands ready, the sequential instruction could be issued if there is no data dependency. In this way, the pipeline stalls can be reduced. In order to issue vector instructions in an out-of-order fashion, OVPM employs two new instruction buffers: a vector arithmetic instruction buffer (VAIB) and a vector memory instruction buffer (VMIB), in the vector datapath as the reorder buffers. Vector instructions in the two buffers would be traversed to detect the states of the operands. When operands of a vector instruction have been prepared, it could be issued no matter whether there are its precedent vector instructions in the program sequence or not.

A simulator of OVPM is developed to evaluate its performance on media workloads. The performance of IVPM and OVPM are compared in order to show the effective of OVPM. As the evaluation results, the performance of OVPM is 3.25x higher than that of IVPM on average. Specially, the performance improvement of MMAs with short vectors is higher than those of MMAs with long vectors. This is because short vector operations cannot be always overlapped with memory operations in the case of IVPM. As a result, they easily expose the pipeline stalls, leading to the low computational efficiency. OVPM could efficiently reduce these stalls, and thereby allow vector extension to achieve a much higher efficiency on processing short vectors. In this chapter, the power consumption of OVMP is also evaluated in order to show its power efficiency. As the evaluation results, OVPM can be implemented with only 7% higher power consumption than IVPM. The additional power consumption of OVPM comes from vector register files, vector reorder buffers, and load and store queues. OVPM adopts a larger number of physical vector register files than IVPM. Those vector register files are used for register renaming in order to omit name dependencies. Moreover, OVPM consumes additional power for realizing the out-of-order execution of vector instructions by using the vector reorder buffers. The load and store queues of OVPM also achieve higher power consumption than those of IVPM because they install new hardware to detect the load-store coherency. These evaluation results confirm that OVPM could effectively improve the power efficiency on the execution of MMAs.

Chapter 3

Conventional multi-banked cache memories for vector architectures cannot achieve high data transfer performance and low power consumption at the same time for MMAs. In order to improve the data transfer performance, conventional vector architectures employ a multi-banked cache memory. For a multi-banked cache memory, cache line size is a key factor to affect the data transfer performance and power consumption. A multi-banked cache memory with small sized cache lines (MBC-S) could achieve a high data transfer performance for various MMAs, while a multi-banked cache memory with large sized cache lines (MBC-L) could achieve low power consumption on its tag arrays. However, MBC-S consumes large power consumption on tag arrays, and MBC-L has inefficiency on short vector processing. Therefore, conventional multi-banked cache memories cannot achieve a high data transfer performance at low power consumption.

In order to achieve a high performance data transfer at low power consumption, a multi-banked cache memory for MVPX, called MVP-cache is proposed in this chapter. Unlike conventional multi-banked cache memories that consist of one data array and one tag array, MVP-cache associates one tag array with

multiple independent data arrays of small-sized cache lines. In this way, MVP-cache realizes less static power consumption on its tag arrays. MVP-cache can also achieve a high efficiency on short vector data transfers because the flexibility of data transfers can be improved by independently controlling the data transfers of each data array.

A simulator of MVP-cache has been developed to evaluate its performance on media workloads. The energy consumption of MBC-S and MBC-L are compared with that of MVP-cache in this chapter. MVP-cache achieves a lower energy consumption for MMAs with short vectors because it can transfer short vector data more efficiently than MBC-L and cost less power on tag arrays than MBC-S. In order to find the energy efficient configuration of MVP-cache, the energy consumption of MVP-cache is evaluated with various numbers of data arrays and sub-caches. The evaluation results show that, with the decrease in the number of data arrays, the energy consumption of the crossbar is also reduced. It achieves a reasonable value in the case of 16 data arrays. At the same time, when the 16 data arrays share one tag array, the energy consumption of the tag array is reduced significantly. Although the average execution cycles of MMAs increase due to stride accesses, the decrease of performance is much smaller than the reduction in energy consumption. Therefore, one tag array associated with 16 data arrays is the most energy efficient configurations for the MMAs. These evaluation results confirm that MVP-cache can achieve a high data transfer performance at low power consumption for MMAs. It is also found that the configuration of 16 data arrays associated with one tag array is a reasonable configuration for media benchmark programs used in this dissertation. When the ranges of MMAs extend, it is also possible to use the same methodology mentioned in this paper to find a reasonable configuration.

Chapter 4

Employing multiple parallel arithmetic pipelines (PAPs) and cache ports (CPs), vector architectures could simultaneously process and transfer the large amount of media, and thus the performance for MMAs could be improved. Therefore, the number of PAPs and the number of CPs could be considered as important parameters to affect the performance of vector architectures. For conventional vector architectures, these two parameters are usually fixed. However, since each MMA has its own hardware requirement, vector architectures cannot always achieve a low energy execution for all MMAs. In order to resolve this problem, the number of PAPs and the number of CPs should be properly configured for each MMA. Therefore, a method to find the lowest energy configuration is required. Moreover, since the lowest energy configuration for an MMA may vary during the execution, it should be dynamically optimized at runtime. Therefore, it is also required to find the lowest energy configuration as quickly as possible.

In order to find the lowest energy configuration as quickly as possible, a performance-power optimization method (PPoM) for MVPX is proposed in this chapter. PPoM adopts the greedy searching method and a performance analytical model based on the enqueue and dequeue throughputs of vector issue queues. This is because these throughputs could reflect the utilization of hardware resources. By using the greedy searching method and the analytical model, PPoM could find the lowest energy configuration more quickly than conventional approaches.

PPoM is integrated with the simulator of MVPX to show its effectiveness. In the evaluation, the energy consumption of vector architecture using the configuration found by PPoM is compared with that of other configurations. As the results, PPoM could find the lowest energy configuration for seven out of nine benchmark programs. Even in the two failed cases, the configurations found by the proposed PPoM can still achieve the second lowest energy consumption. PPoM is failed for two benchmark programs because there is no significant performance improvement unless the numbers of CPs and PAPs are increased at the same time. Such a situation occurs because the computing performance and data transfer performance are well balanced in a certain configuration. However, since PPoM can only increase one kind of hardware resources each time, it fails to find the lowest energy consumption for the two benchmark programs.

Chapter 5

In order to achieve a high power-efficient execution on MMAs, MVPX is proposed in this dissertation. To overcome the inefficiency of conventional vector architectures for MMAs, an out-of-order vector mechanism, a multi-banked cache memory that associates one tag array with multiple data arrays and a performance-power optimization method have been proposed in this dissertation. These proposed solutions have solved the issues on different layers of the existing vector architecture for MMAs. Enhanced with these proposed solutions, it is possible for MVPX to accelerate a wide range of MMAs with high power efficiency.

論文審査結果の要旨

近年，音声，画像，映像等を扱うマルチメディアアプリケーションの高度化が進み，それらを汎用マイクロプロセッサで高性能，かつ低消費電力で実行することが困難になりつつある。このため，これらのアプリケーションの有するデータ並列性を効率よく利用し，高い電力効率で高性能なマルチメディア用マイクロプロセッサの実現が求められている。本論文は，データ並列処理性能を高めるためのベクトル拡張命令セットを汎用マイクロプロセッサ向けに定義し，ベクトル拡張命令セットを高スループット，かつ低消費電力で処理できるベクトルアーキテクチャの設計とその最適化手法を論じたものであり，全編5章からなる。

第1章は緒論である。

第2章では，まず，ベクトル拡張命令により様々なマルチメディアアプリケーションに内在するデータ並列性を効果的に抽出できることを明らかにしている。さらに，アウトオブオーダ型のベクトル処理機構を提案し，マルチメディアアプリケーションに数多く含まれる短ベクトル処理で生じていたメモリストールを解決し，実効性能の向上と電力効率の改善が得られることを評価実験により定量的に明らかにしている。本章で得られた知見は，高い電力効率を実現する次世代のベクトルアーキテクチャ設計において有用である。

第3章では，高いデータ転送性能と低消費電力を両立するベクトルアーキテクチャのためのマルチバンクキャッシュ機構を提案している。従来のベクトル処理向けのマルチバンクキャッシュにおいて多大な消費電力を必要としていたタグアレイ管理を効率化し，かつ多様なベクトル長のデータ転送にも柔軟に対応可能にさせることにより，高いデータ転送能力を維持したまま，回路規模と消費電力の大幅な削減を達成している。本章で得られた知見は，将来の高電力効率で高バンド幅なキャッシュの設計において重要な指針を与えている。

第4章では，ベクトル拡張命令を有するベクトルアーキテクチャの電力効率を最大化するためのアーキテクチャ最適化手法を提案している。まず，最も高い電力効率を達成可能なプロセッサの構成は，マルチメディアアプリケーションによって異なることを見いだし，次に，アプリケーション毎に電力効率の優れた構成を，性能モデルを用いた設計空間探索で導き出すアーキテクチャ最適化手法を提案している。本章で得られた知見は，高性能，かつ低消費電力なマルチメディア向けベクトルプロセッサのアプリケーション毎に最適設計する上で，有益な成果である。

第5章は，本論文を総括し，結論としている。

以上要するに本論文は，次世代マルチメディアアプリケーションのためのベクトルアーキテクチャ設計において，高性能と低消費電力を両立させるための重要な知見を与えたものであり，情報基礎科学ならびに計算機科学の発展に寄与するところが少なくない。

よって，本論文は博士（情報科学）の学位論文として合格と認める。